COURSE STRUCTURE

For

B.TECH. DEGREE

in

COMPUTER SCIENCE & ENGINEERING

(Applicable from the academic session 2024-2025)

Dr. B. C. Roy Engineering College

An Autonomous Institution

Approved by: All India Council for Technical Education (AICTE)
Affiliated to: Maulana Abul Kalam Azad University of Technology, West Bengal
(Formerly Known as -WBUT)

Jemua Road, Durgapur, West Bengal, India, 713206

- The first year course structure (Page 3 and Page 4) is unanimously accepted and approved in the first BoS meeting held in the Department of a) Physics, b) Chemistry, c) Mathematics, d) English, e) Electrical Engineering, f) Electronics and Communication Engineering, g) Computer Science and Engineering, h) Mechanical Engineering.
- The BoS of CSE (Computer Science and Engineering) in its first meeting (held in the Department of CSE (Computer Science and Engineering) on 6th November 2024 has unanimously accepted and approved the four year course structure of CSE (Computer Science and Engineering).

Head
Dept. Computer Science & Eng
Dr. B. C. Roy Engineering College
Durgapur

Semester 4								
Sl No	**Course Type**	**Paper Code**	**Paper Name**	**Marks**	**L**	**T**	**P**	**Credit**
Theory								
1	PC	CS401	Design & Analysis of Algorithms	100	3	0	0	3
2	PC	CS402	Object Oriented Programming	100	3	0	0	3
3	PC	CS403	Database Management System	100	3	0	0	3
4	PC	CS404	Formal Language & Automata Theory	100	3	0	0	3
5	BS	CS405	Discrete Mathematics & Graph Theory	100	3	0	0	3
6	HM	CS406	Soft Skill & Interpersonal Communication	100	3	0	0	3
			Total Theory	600	18	0	0	18
Practical								
7	PC	CS491	Design & Analysis of Algorithms Lab	100	0	0	4	2
8	PC	CS492	Object Oriented Programming Lab	100	0	0	4	2
9	PC	CS493	Database Management System Lab	100	0	0	4	2
			Total Practical	300	0	0	12	6
			Total of 4th Semester	900	18	0	12	24
Total Credits								**24**

Course Name: Design and Analysis of Algorithms
Course Code: CS401
(Semester – IV)
Category: Major
Course Broad Category: Engineering Science Course
[For Computer Science and Engineering]

1. **Course Prerequisite:**
   Arrays, Linked Lists, Stacks, Queues, Tree data structure (Binary Trees, BSTs, Heaps), Proficiency in a programming language

2. **Course Learning Objectives:**
   i. The aim of this module is to learn how to develop efficient algorithms for simple computational tasks and reasoning about the correctness of them
   ii. Through the complexity measures, different range of behaviors of algorithms and the notion of tractable and intractable problems will be understood.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies.
   Evaluation System –
   A. Mid-Term Exam (20 Marks)- Summative Assessment (CIA-1)
   B. Internal Assessment (20 Marks)- Formative Continuous Assessment [Continuous Assessment 1 (CIA-2)]
   C. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**
   Course Name: Design and Analysis of Algorithms
   Course Code: CS401
   Hours per Week: 3L: 0T: 0P
   Credits: 3

Module	Topics	Lectures
Unit 1: Introduction	Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem	6L
Unit 2: Fundamental Algorithmic Strategies	Brute-Force (Conceptualization in String Matching, Closest Pair Problem, TSP using Exhaustive Search), Greedy(Fractional Knapsack, Activity Selection, Huffman Coding), Dynamic Programming (0/1 Knapsack, LCS, Matrix Chain Multiplication, TSP), Branch and- Bound (Bounding Function, TSP , 0/1 Knapsack) and Backtracking (State-space Search, N-Queens Problem, Hamiltonian Cycle) methodologies for the design of algorithms; Heuristics –characteristics and their application domains (Characteristics, Approximate Solutions, 15-puzzle problem)	10L
Unit 3: Graph and Tree Algorithms	Graph and Tree Traversal Algorithms (Depth First Search , Breadth First Search); Shortest Path Algorithms (Dijkstra's Algorithm, Bellman-Ford Algorithm, Floyd-Warshall Algorithm); Minimum Spanning Tree (Prim's Algorithm, Kruskal's Algorithm); Topological Sorting (DFS-based Sorting); Network Flow Algorithm (Ford-Fulkerson Method).	7L
Unit 4: Tractable and Intractable Problems	Computability of Algorithms [1L] : Definition of computability and decidability, Turing Machines as a model of computation, Decidable vs. Undecidable Problems (Halting Problem) Computational Complexity and Complexity Classes [2L]: Complexity Classes: P, NP, NP-Complete, and NP-Hard, Relationships between P, NP, NP-Complete, and NP-Hard Cook's Theorem and NP-Completeness [2L] : Statement and significance of Cook's Theorem, SAT (Boolean Satisfiability Problem) as NP-Complete, Concept of polynomial-time reductions Standard NP-Complete Problems [2L]: Satisfiability Problems (SAT, 3-SAT, k-SAT), Graph Problems (Hamiltonian Cycle, Clique Problem, Vertex Cover), Subset and Partition Problems (Knapsack Problem, Subset Sum Problem), Traveling Salesman Problem (TSP) as NP-Hard Reduction Techniques in Complexity Theory [2L]: Concept of reductions (Polynomial-time reductions, Mapping reductions), Examples of problem reductions (Reducing 3-SAT to Clique, Reducing Vertex Cover to Independent Set), Role of reductions in proving NP-completeness	10L
Unit 5: Advanced Topics	Approximation Algorithms [1L]: Need for Approximation Algorithms (NP-hard problems, In approximability), Performance Ratio and Approximation Bounds, Vertex Cover (Greedy Approach, Factor-2 Approximation), Traveling Salesman Problem (TSP) – Metric Approximation using MST, Knapsack Problem – FPTAS (Fully Polynomial-Time Approximation Scheme) Randomized Algorithms [1L]: Introduction to Randomized Algorithms (Las Vegas vs Monte Carlo algorithms), Applications of Randomization in Algorithms	5L

6. **References:**

Text & References Books:

- Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
- Fundamentals of Algorithms – E. Horowitz et al.
- Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
- Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
- Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA
- Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House (AICTE Recommended Textbook – 2018)
- Algorithms Design and Analysis, Udit Agarwal, Dhanpat Rai

7. Course Outcomes:

After going through this course the Students will be able to:

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS401.1	Understand and apply asymptotic notations to evaluate the time and space complexity of both recursive and non-recursive algorithms.	Analyze	L4
CS401.2	Design and implement solutions for optimization problems using Greedy and Dynamic Programming approaches while evaluating their efficiency.	Develop, Evaluate	L5
CS401.3	Utilize Backtracking and Branch-and-Bound strategies to solve complex computational problems..	Apply	L3
CS401.4	Construct graph representations and apply traversal, shortest path, and spanning tree algorithms for real-world problem-solving.	Implement	L6
CS401.5	Explore NP-completeness concepts and apply approximation and randomized algorithms to tackle intractable problems.	Understand, Apply	L2, L3
CS401.6	Analyze different algorithmic approaches and select the most appropriate one for ensuring computational efficiency in diverse application domains.	Evaluate	L5

8. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1	3	-	-	-	-	-
2	-	3	3	-	-	-
3	-	-	-	3	-	-
4	-	-	-	-	3	-
5	-	-	-	-	-	3

9. Mapping of the Course outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	-	-	-	-	-	-	-	2
CO2	3	3	3	2	2	-	-	-	2	-	-
CO3	2	3	3	3	-	-	-	1	-	-	-
CO4	3	2	3	3	3	2	-	-	2	2	-
CO5	2	2	2	3	3	3	2	-	-	3	2
CO6	3	3	3	-	3	3	3	3	-	2	3
AVG.	**2.7**	**2.7**	**2.7**	**2.7**	**2.7**	**2.7**	**2.5**	**2**	**2.0**	**2.3**	**2.3**

10. Mapping to PSO

Course Outcome	PSO1	PSO2	PSO3
CS401.CO1	3	—	—
CS401.CO2	3	2	1
CS401.CO3	3	1	—
CS401.CO4	3	1	1
CS401.CO5	3	—	2
CS401.CO6	3	2	1

***** End of Syllabus*****

Course Name: Object Oriented Programming
Course Code: CS402
(Semester – IV)
Category: Major
Course Broad Category: Engineering Science Course
[For Computer Science and Engineering]

……

1. **Course Prerequisite:**
   - **Basic knowledge of programming** (preferably in C or any procedural language).
   - **Understanding of data structures, including arrays, linked lists, stacks, and queues.**
   - **Familiarity with algorithmic problem-solving and logic development.**
   - **Basic knowledge of memory management concepts.**

2. **Course Learning Objectives:**
   After completing this course, students will be able to:
   i. Understand the principles of object-oriented programming and design.
   ii. Apply Java programming concepts for real-world problem solving.
   iii. Implement object-oriented features like inheritance, polymorphism, encapsulation, and abstraction.
   iv. Use Java's I/O operations and String handling mechanisms efficiently.
   v. Handle exceptions and implement multithreading in Java.
   vi. Create GUI and applet-based applications.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies.
   Evaluation System –
   A. Mid-Term Exam (20 Marks)- Summative Assessment (CIA-1)
   B. Internal Assessment (20 Marks)- Formative Continuous Assessment [Continuous Assessment 1 (CIA-2)]
   C. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**
   Course Name: Object Oriented Programming
   Course Code: CS402
   Hours per Week: 3L: 0T: 0P
   Credits: 3

Module	Topics	Lectures
Unit 1: Concepts of Object-Oriented Programming	Object Oriented Analysis (OOA) & Object Oriented Design (OOD) - Concepts of object oriented programming language, difference between Procedural and Object Oriented Language, Various Object Oriented Programming (e.g. C++, Java, and Python etc.) [1L]; Relationships among objects and classes-generalization, Specialization, Aggregation, Association, Composition, links, Meta-class. [1L];	2L
Unit 2: Java Basics:	Basic concepts of java programming – Various features of java, Byte-code & JVM, Data types, Different types of Variables.[1L] ;Java Operators & Control statements [1L]; Java loops. [1L]; Array. [1L] ;Creation of class, object, method. [1L]; Constructor- Definition, Usage of Constructor, Different types of Constructor. [1L]; finalize method and garbage collection, Method & Constructor overloading. [1L]; this keyword, use of objects as parameter & methods returning objects. [1L]; Call by value & call by reference. [1L]; Static variables & methods. Nested & inner classes. [1L].	10L
Unit 3: Basic String handling & I/O	Basic string handling concepts- Concept of mutable and immutable string, Methods of String classcharAt(), compareTo(), equals(), equalsIgnoreCase(), indexOf(), length() , substring(). [1L]; to CharArray(), toLowerCase(), toString(), toUpperCase() , trim() , valueOf() methods, Methods of String buffer class- append(), capacity(), charAt(), delete(), deleteCharAt(). [1L];ensureCapacity(), getChars(), indexOf(), insert(), length(), setCharAt(), setLength(), substring(), toString(). [1L]; Command line arguments, basics of I/O operations – keyboard input using BufferedReader [1L] ; Scanner class in Java I/O operation [1L];	5L

Unit 4: Inheritance and Java Packages:	Inheritance - Definition, Advantages, Different types of inheritance and their implementation. [1L] ;Super and final keywords, super() method. [1L]; Method overriding, Dynamic method dispatch.[1L]; Abstract classes & methods.[1L]; Interface - Definition, Use of Interface.[1L]; Multiple inheritance by using Interface.[1L] ;Java Packages -Definition, Creation of packages. [1L]; Java Access Modifiers - public, private, default and protected, Importing packages, member access for packages. [1L]	8L
Unit 5: Exception handling, Multithreading, Applet Programming and Swing :	Exception handling - Basics, different types of exception classes. Difference between Checked & Unchecked Exception.[1L]; Try & catch related case studies.[1L]; Throw, throws & finally. [1L]; Creation of user defined exception. [1L]; Multithreading - Basics, main thread [1L]; Thread life cycle. [1L]; Creation of multiple threads-yield(), suspend(), sleep(n), resume(), wait(), notify(), join(), isAlive(). [1L] ;Thread priorities, thread synchronization.[1L]; Inter thread communication, deadlocks for threads[1L]; Applet Programming - Basics, applet life cycle, difference between application & applet programming[1L]; Parameter passing in applets, Swings[1L]	11L

6. References:

Text & References Books:

- Herbert Schildt – "Java: The Complete Reference " – TMH
- E. Balagurusamy– " Programming With Java: A Primer"–TMH
- R.K Das– "Core Java For Beginners"– VIKASPUBLISHING
- Rambaugh, James Michael, Blaha– "Object Oriented Modelling and Design"– Prentice Hall, India
- Patrick Naughton, Herbert Schildt– "The complete reference-Java2"– TMH
- Deitel and Deitel– "Java How to Program".– Pearson

7. Course Outcomes:

After going through this course the Students will be able to:

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS402.1	**Explain** the concepts of Object Oriented Analysis and Design and compare Java with other OOP languages..	Understand	L2
CS402.2	**Apply** Java fundamentals, loops, arrays, constructors, and class objects in problem-solving..	Apply	L3
CS402.3	**Demonstrate** proficiency in string handling and basic I/O operations in Java..	Apply	L3
CS402.4	**Develop** modular Java programs using inheritance, interfaces, and packages.	Create	L6
CS402.5	**Construct** robust applications using exception handling and multithreading.	Create	L6
CS402.6	**Design** interactive programs using Applet and GUI elements.	Create	L6

8. Mapping of course outcomes to module / course content

Module	Topics Covered	Mapped COs
1	OOA, OOD, OOP basics, Java vs C++	CO1
2	Java basics, class, object, constructors, this keyword, static, inner class	CO2
3	String handling, BufferedReader, Scanner	CO3
4	Inheritance, Interface, Packages	CO4
5	Exception Handling, Multithreading, Applet	CO5, CO6

9. Mapping of the Course outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	1	1	1	1	-	-	1	-	1
CO2	3	3	2	2	1	-	-	1	2	-	1
CO3	2	2	2	2	2	-	-	-	1	-	1
CO4	3	2	3	2	2	-	-	1	1	-	1
CO5	3	2	3	2	3	-	-	1	2	-	1
CO6	3	2	2	1	2	-	-	1	2	-	2
AVG.	**3**	**2.16**	**2.16**	**1.66**	**1.83**	**1**	**-**	**1**	**1.50**	**-**	**1.16**

10. Mapping to PSO

Course Outcome	PSO1	PSO2	PSO3
CS402.CO1	1	2	—
CS402.CO2	2	3	—
CS402.CO3	1	2	—
CS402.CO4	2	3	—
CS402.CO5	2	3	1
CS402.CO6	1	2	—

***** End of Syllabus*****

Course Name: Database Management System
Course Code: CS403
(Semester – IV)
Category: Major
Course Broad Category: Engineering Science Course
[For Computer Science and Engineering]

……

1. **Course Prerequisite:** Fundamental computer knowledge that includes concepts of computer architecture, storage and hardware. Knowledge of data structures and algorithms and programming will be an added benefit. …

2. **Course Learning Objectives:**
   i. To understand the different issues involved in the design and implementation of a database system.
   ii. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
   iii. To understand and use data manipulation language to query, update, and manage a database
   iv. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
   v. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.
   vi. To understand the different issues involved in the design and implementation of a database system.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies.
   Evaluation System –
   A. Mid-Term Exam (20 Marks)- Summative Assessment (CIA-1)
   B. Internal Assessment (20 Marks)- Formative Continuous Assessment [Continuous Assessment 1 (CIA-2)]
   C. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**
   Course Name: Database Management System
   Course Code: CS403
   Hours per Week: 3L: 0T: 0P
   Credits: 3

Module	Topics	Lectures
Unit 1:	**Database System Architecture –** Data Abstraction (physical, logical, view levels), Data Independence, Database Languages: DDL and DML*(2 Lectures)* **Data Models Overview –**Hierarchical, Network, Relational, Object-Oriented Models Introduction to Logical Data Models **(**1 Lecture**)** **Relational Data Model –**Structure of Relational Databases (tables, attributes, tuples),Keys: Superkey, Candidate Key, Primary Key, Foreign Key, Integrity Constraints: Domain, Entity, Referential *(2 Lectures)* **Introduction to RDBMS –** Characteristics and Components, Differences from DBMS, Commercial examples (Oracle, MySQL, PostgreSQL)*(1 Lecture)* **Entity-Relationship (ER) Modelling –**Entities, Attributes, Relationships, ER Diagram Notations (Chen/Min-Max) *(1 Lecture)* **Database Design Process –**Conceptual → Logical → Physical Design, Importance of good design *(1 Lecture)* **Mapping from ER Model to Relational Model –**Step-by-step mapping of entity sets, relationships, constraints*(2 Lectures)*	10L
Unit 2:	**Relational query languages:** Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server. **Relational database design:** Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design. **Query processing and optimization:** Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.	12L
Unit 3:	**Storage strategies:** Indices, B-trees, hashing.	5L
Unit 4:	**Transaction processing:** Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi- version and optimistic Concurrency Control schemes, Database recovery.	6L
Unit 5:	**Database Security:** Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.	3L

Unit 6:	**Advanced topics:** Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Unstructured Databases and NoSQL, Concept of unstructured data (text, images, audio, etc.).	4L

6. References:

Text & References Books:

- "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
- "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
- Database Management Systems, R.P. Mahapatra, Khanna Publishing House, New Delhi (AICTE Recommended Textbook – 2018)
- "Fundamentals of Database Systems" , 5th Edition by R. Elmasri and S. Navathe,
- PearsonEducation "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley.

7. Course Outcomes:

After going through this course the Students will be able to:

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS403.1	Describe the basic concept of database and different database models along with database languages like DDL, DML etc, Data Abstraction, and Data Independence.	Describe	L1
CS403.2	Identify different approaches for solving queries such as Relational algebra, Tuple and domain relational calculus, considering the query optimization strategies, and different normal forms for relational database normalization.	Identify	L3
CS403.3	Evaluate the applications of different storage strategies such as Indices, B-trees, hashing	Evaluate	L5
CS403.4	Understand the transaction processing and concurrency control strategies including ACID property, serializability of scheduling, locking and timestamp based schedulers, Database recovery.	Understand	L2
CS403.5	Analyze the database security approaches including authentication, authorization and access control, DAC, MAC and RBAC models, intrusion detection, SQL injection etc.	Analyze	L4
CS403.6	Explain the advanced concepts related to DBMS such as object oriented and object relational databases, logical databases, web databases, distributed databases, data warehousing and data mining.	Explain	L2

8. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1	Y	Y	-	-	-	-
2	-	Y	-	-	-	-
3	-	-	Y	-	-	-
4	-	-	-	Y	Y	-
5	-	-	-	-	Y	-
6	-	-	-	-	-	Y

9. Mapping of the Course outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	2	2	1	1	-	-	-	-	-	1
CO2	2	-	2	2	-	1	-	-	-	-	-
CO3	-	2	2	2	2	-	2	-	-	1	-
CO4	1	3	3	1	2	-	-	-	-	-	-
CO5	1	2	-	2	1	-	-	-	-	2	-
CO6	2	2	2	2	2	3	-	-	-	-	-
AVG.	**1.6**	**2.2**	**2.2**	**1.6**	**1.6**	**2.0**	**2.0**	**0.0**	**0.0**	**1.5**	**1.0**

10. Mapping to PSO

Course Outcome	PSO1	PSO2	PSO3
CS403.CO1	1	1	—
CS403.CO2	3	2	—
CS403.CO3	2	2	—
CS403.CO4	2	2	1
CS403.CO5	2	2	1
CS403.CO6	2	2	2

***** End of Syllabus*****

Course Name: Formal Language & Automata Theory
Course Code: CS404
(Semester – IV)
Category: Major
Course Broad Category: Engineering Science Course
[For Computer Science and Engineering]

……

1. **Course Prerequisite:**
   Basic Knowledge of Set theory, Graph Theory, and Functions & Relations.

2. **Course Learning Objectives:**
   i. Be able to construct finite state machines and equivalent regular expressions.
   ii. Be able to prove the equivalence of languages described by finite state machines and regular expressions
   iii. Be able to construct pushdown automata, the equivalent context-free grammars
   iv. Be able to construct Turing machines and Post machines. Be able to prove the equivalence of languages described by Turing machines and Post machines.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions, and Case Studies.
   Evaluation System –
   A. Mid-Term Exam (20 Marks)- Summative Assessment (CIA-1)
   B. Internal Assessment (20 Marks)- Formative Continuous Assessment [Continuous Assessment 1 (CIA-2)]
   C. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**
   Course Name: Formal Language & Automata Theory
   Course Code: CS404
   Hours per Week: 3L: 0T: 0P
   Credits: 3

Module	Topics	Lectures
Unit 1: Introduction	Alphabet, languages and grammar, productions and derivation, Chomsky hierarchy of languages.	7L
Unit 2: Regular languages and finite automata	Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata.	7L
Unit 3: Context-free languages and pushdown automata	Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic push-down automata, closure properties of CFLs.	6L
Unit 4: Context-sensitive languages	Context-sensitive grammars (CSG) and Context-sensitive languages, linear bounded automata, and equivalence with CSG.	6L
Unit 5: Turing machines	The basic model for Turing machines (TM), Turing recognizable(recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators	6L
Unit 6: Undecidability	Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages	6L

6. **References:**
   Text & References Books:

- John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.
- K.L.P. Mishra, N. Chandrasekaran, Theory of Computer Science.
- Peter Linz, An Introduction to Formal Languagesand Automata
- Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.

- John Martin, Introduction to Languages and The Theory of Computation, TataMcGraw Hill., PEARSON.

7. Course Outcomes:

After going through this course, the students will be able to:

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS404.1	Understand formal notations for strings, languages, and machines.	Understand	L2
CS404.2	Design finite automata to accept a set of strings of a language and to determine whether the given language is non-regular.	Create	L6
CS404.3	Design context-free grammar to generate context-free language and Push down Automata.	Apply	L3
CS404.4	Understand Context-sensitive grammars (CSG) and languages, linear bounded automata, and equivalence with CSG.	Analyze	L4
CS404.5	Understand the basic model of Turing machine(TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages.	Understand	L2
CS404.6	Distinguish between computability and non-computability, Decidability and undecidability.	Analyze	L4

8. Mapping of course outcomes to module/course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1	✓	-	-	-	-	-
2	-	✓	-	-	-	-
3	-	-	✓	-	-	-
4	-	-	-	✓	-	-
5	-	-	-	-	✓	-
6	-	-	-	-	-	✓

9. Mapping of the Course Outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	3	2	-	-	-	-	-	-	3
CO2	3	3	3	2	-	-	-	-	-	-	3
CO3	3	3	3	2	-	-	-	-	-	-	3
CO4	3	3	3	2	-	-	-	-	-	-	3
CO5	3	3	3	2	-	-	-	-	-	-	3
CO6	3	2	3	2	-	-	-	-	-	-	3
AVG.	**3**	**2.66**	**3**	**2**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**3**

10. Mapping to PSO

Course Outcome	PSO1	PSO2	PSO3
CS404.CO1	2	—	—
CS404.CO2	3	—	—
CS404.CO3	3	—	—
CS404.CO4	2	—	—
CS404.CO5	2	—	—
CS404.CO6	3	—	—

***** End of Syllabus*****

Course Name: Discrete Mathematics&Graph theory
Course Code: CS405
(Semester IV)
Course Broad Category: B. Tech CSE

1. **Course Prerequisite:**
   Concept of Mathematics in the previous semesters of B. Tech curriculum.

2. **Course Learning Objectives:**
   Introduces the elementary discrete mathematics for computer science and engineering.
   Topics include formal logic notation, methods of proof, induction, sets, relations, graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies, Guest Lectures.

   Evaluation System –

A. Mid-Term Exam (20 Marks)- Summative Assessment (CIA-1)
B. Internal Assessment (20 Marks)- Formative Continuous Assessment [Continuous Assessment 1 (CIA-2)]
C. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**
   Course Name: Discrete Mathematics and graph theory
   Course Code: CS405(CSE)
   Hours per Week: 3L: 1T: 0P
   Credits: 3

Module	Topics	45L
1.	**Mathematical Logic and Relations**: Statements and notations, Connectives, Well-formed formulas, Truth Tables, tautology, equivalence implication, Normal forms, Quantifiers, universal quantifiers. Properties of binary relations, equivalence relations, partial ordering relations, Lattices, Hasse diagram. **Predicates**: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving.	6L
2.	**Algebraic structures:** Algebraic systems, examples and general properties, Semigroups, Monoids and Groups, different Finite and Infinite groups. **Elementary Combinatorics:** Basis of counting, Sum rule and Product rule, Combinations & Permutations, Binomial Coefficients, Binomial Multinomial theorems, the principles of Inclusion – Exclusion. Pigeon hole principles and its application.	10L
3.	**Recurrence Relation:** Generating Functions, Calculating Coefficient of generating function, Recurrence relations, solving recurrence relation by substitution and Generating functions, Characteristics roots solution of In-homogeneous Recurrence Relation.	5L

4.	**Basic Graph Theory:** **Basic Concept of graph theory**- Definitions and Matrix Representation: Incidence & Adjacency matrix. Weighted graph, Connected and disconnected graphs, complement of a graph, Regular graph, Complete graph, Sub-graph, Walk, Path, Circuit, Euler and Hamiltonian graph- Necessary conditions and sufficient conditions, di-graph, cut sets and cut vertices, Graph isomorphism, Bipartite Graphs- properties of Bipartite Graphs, Dijkstra's Algorithm for shortest path problem. **Graph Operations:** Union, Sum, Cartesian Product, Composition, Graphic sequences, Havel- Hakimi criterion, Realization of a graphic sequence.	10L
5.	**Advanced Graph Theory:** **Tree:**Definitions and characterizations, Number of trees, Cayley's formula, Kirchoff-matrix-tree theorem, **Spanning tree:** Definition, Number of minimum spanning trees, BFS and DFS algorithms, Minimal spanning tree- Kruskal's and Prim's algorithms. Eulerian Graphs,Fleury's algorithm, Chinese Postman problem, Hamilton Graphs, Introduction, Necessary conditions and sufficient conditions. **Planar graphs-** Definitions and theorems, Kuratowski's theorem for testing planarity, Duality in planar graphs. **Graph coloring:**Vertex Colorings- Basic definitions, Cliques and chromatic number, Greedy coloring algorithm, Brooks theorem, Edge Colorings. **Independent sets coverings and matchings:** - Introduction, Independent sets and coverings: basic equations, Matchings in bipartite graphs, Hall's Marriage Theorem (Statement only), K¨onig's Theorem (Statement only), Perfect matchings in graphs, Greedy and approximation algorithms.	14L

5. References:

Text Book:

- Rosen, Kenneth H.: Discrete Mathematics and its Applications with Combinatorics and Graph Theory (7th Edition), TMH (Tata McGraw-Hill).
- Mott, Joe L., Kandel, Abraham, and Baker, Teodore P.: Discrete Mathematics for Computer Scientists and Mathematicians, Pearson Education.
- Johnsonbaugh, Richard: Discrete Mathematics, Pearson Education.
- Chandrasekaran, N., and Umaparvathi, M.: Discrete Mathematics, PHI Learning.
- J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, Macmillan Press.

Reference Books:

- Tremblay, J.P., and Manohar, R.: Discrete Mathematical Structures with Applications to Computer Science, TMH (Tata McGraw-Hill).
- Mott, Joe L., Kandel, Abraham, and Baker, Teodore P.: Discrete Mathematics for Computer Scientists and Mathematicians (2nd Edition), Pearson Education.
- Johnsonbaugh, Richard: Discrete Mathematics (7th Edition), Pearson Education.
- Goodaire, Edgar G., and Parmenter, Michael M.: Discrete Mathematics with Graph Theory, Pearson Education.
- Grimaldi, Ralph P.: Discrete and Combinatorial Mathematics: An Applied Introduction (5th Edition), Pearson Education.
- D.B. West, Introduction to Graph Theory, Pearson Education.
- K.R. Parthasarathy, Basic Graph Theory, Tata McGraw Hill Education.
- Sudarsan Nanda, Graph Theory and Algorithms, Allied Publishers.

- M. Mukundarajan, Graph Theory with Applications to Computer Science and Engineering, Yes Dee Publishing.

6. Course Outcomes (CO):

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS405.1	Ability to understand and construct precise mathematical proofs.	Identify	Remember
CS405.2	Ability to use logic and set theory to formulate precise statements.	Explain	Understand
CS405.3	Ability to analyze and solve counting problems on finite and discrete structures	Implement	Apply
CS405.4	Ability to apply graph theory in solving problems in computer science and information technology.	Organize	Analyze
CS405.5	Ability to apply the concept of advanced graph theory in networking and shortest path problems.	Assess	Evaluate
CS405.6	Build up logical and analytical skills to create a new idea appreciated by academics, research & emerging trends in industry.	Construct	Create

7. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1	3	2	1	-	-	1
2	3	3	2	-	-	1
3	1	1	3	-	-	1
4	2	1	2	3	-	1
5	2	1	1	3	3	1

8. Mapping of the Course outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	2	1	1	2	-	-	-	-	-	1
CO2	1	2	3	1	-	-	-	-	-	-	1
CO3	1	2	2	1	1	-	-	-	-	-	1
CO4	1	2	1	1	2	-	-	-	-	-	2
CO5	2	2	2	2	3	-	-	-	-	-	1
CO6	1	1	1	1	-	-	-	-	-	-	1

9. Mapping to Program Specific Outcome (PSO)

Course Outcome	PSO1	PSO2	PSO3
CS405.CO1	2	—	—
CS405.CO2	2	—	—
CS405.CO3	3	—	—
CS405.CO4	3	1	1
CS405.CO5	3	1	1
CS405.CO6	2	—	2

Course Name: Design and Analysis of Algorithms Lab
Course Code: CS491
(Semester – IV)
Category: Major
Course Broad Category: Engineering Science Course
[For Computer Science and Engineering]

1. **Course Prerequisite:**
   To know data structure and basic programming ability

2. **Course Learning Objectives:**
   i. The aim of this module is to learn how to develop efficient algorithms for simple computational tasks and reasoning about the correctness of them
   ii. Through the complexity measures, different ranges of behaviors of algorithms and the notion of tractable and intractable problems will be understood.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies.
   Evaluation System –
   For each laboratory courses there shall be a CIA during the semester for 60 marks and 40 marks for ESE.
   - **Continuous Internal Assessment of laboratory courses (PCIA) [60 Marks]**
     - Regular Laboratory Performance (20 Marks)
     - Experiment Execution and Results (15 Marks)
     - Laboratory Record/Report (10 Marks)
     - Viva Voce/Quiz (5 Marks)
     - Attendance (10 Marks)
   - **End Semester Examination of Laboratory Course (PESE) [40 marks]**
     - Write-Up about the given Experiment/Program [10 marks]
     - Execution of the Experiment/Program [10 marks]
     - Evaluation of the outcome or results of the Experiment/Program [10 marks]
     - Viva-Voce (on the concerned Experiment/Programming) [10 marks]

4. **Course Content:**
   Course Name: Design and Analysis of Algorithms Lab
   Course Code: CS491
   Hours per Week: 4P
   Credits: 2

Unit 1: Divide and conquer,	● Sort a given set of elements using the Quicksort algorithm and determine the time required to sort the elements. Repeat the experiment for different values of n. The elements can be read from a file or can be generated using the random number generator. ● Sort a given set of elements using the merge sort algorithm and determine the time required to sort the elements. Repeat the experiment for different values of n. The elements can be read from a file or can be generated using the random number generator.
Unit 2: Greedy method	● Imagine a situation where you are a thief breaking into a store that has n valuable items. Each item has a specific weight w[i] and profit p[i] (or value). However, your knapsack has a maximum capacity W, and you can't carry more than that weight. Your task is to determine the optimal set of items (or item portions) to carry so that the total profit is maximized, without exceeding the weight limit of the knapsack. You are required to solve this using a Greedy Algorithm.

<table>
<tr><td></td><td>

Imagine you are a freelance worker who receives n job offers from different clients. Each job j[i] takes 1 unit of time to complete and offers a profit p[i]. However, each job also has a deadline d[i], which means it must be completed on or before time d[i] to earn the profit. You can only do one job at a time, and your goal is to maximize your total profit by selecting and scheduling the most profitable jobs within their respective deadlines. Your task is to determine the optimal sequence of jobs to undertake using a Greedy Algorithm so that the total profit is maximized, while ensuring that no deadlines are missed.
In real-world applications like Google Maps, network routing, or robot navigation, one common problem is to find the shortest path between two points in a network. Given a graph and a source vertex, the goal is to find the shortest path (minimum total weight) from the source to all other vertices in the graph.

<table>
<tr><th>From</th><th>To</th><th>Weight</th></tr>
<tr><td>A</td><td>B</td><td>6</td></tr>
<tr><td>A</td><td>D</td><td>1</td></tr>
<tr><td>D</td><td>B</td><td>2</td></tr>
<tr><td>B</td><td>E</td><td>2</td></tr>
<tr><td>D</td><td>E</td><td>1</td></tr>
<tr><td>E</td><td>C</td><td>5</td></tr>
</table>

You have a list of m potential roads that can be built, with each road connecting two cities and having a specific construction cost. Your goal is to build a road network that connects all the cities together while minimizing the total construction cost. The roads are to be chosen in such a way that every city is connected to every other city, but no city is connected to itself, and there are no redundant roads.

</td></tr>
<tr><td>Unit 3:
Dynamic programming</td><td>

Implement matrix chain multiplication problem using dynamic programming.
Imagine a salesperson who must visit a list of cities exactly once and then return to the starting city, forming a complete tour. The goal is to find the shortest possible path that allows this round-trip. Each pair of cities has a distance (or cost) associated with traveling between them.
You're at a bank or vending machine, and you need to make change for a given amount of money using the fewest number of coins. You are given a list of coin denominations (e.g., ₹1, ₹2, ₹5, ₹10), and you can use an unlimited number of each. Your task is to determine the minimum number of coins required to make a specific amount, or report that it is not possible with the given coin types.
Given two sequences (strings, arrays, etc.), your task is to find the length of their Longest Common Subsequence (LCS). A subsequence is a sequence that appears in the same relative order but not necessarily contiguous. The goal is to identify the longest sequence that appears in both input strings.

</td></tr>
<tr><td>Unit 4:
Graph traversal algorithm</td><td>

Implement BFS and DFS for a given graph.

</td></tr>
<tr><td>Unit 5
Brunch and Bound, backtracking</td><td>

Imagine you are the manager of a chess tournament and need to arrange n queens on an n×n chessboard. The queens must be placed in such a way that no two queens threaten each other. This means that no two queens can share the same row, column, or diagonal. Your task is to determine all possible ways to place the queens on the board such that no two queens can attack each other. You are required to solve this using a Backtracking Algorithm, which efficiently explores all possible placements and ensures the solution meets the required constraints.

</td></tr>
</table>

Any experiment specially designed by the college (Detailed instructions for Laboratory Manual to be followed for further guidance)

5. References:

Text & References Books:

- Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
- Fundamentals of Algorithms – E. Horowitz et al.
- Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
- Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
- Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA
- Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House (AICTE Recommended Textbook – 2018)
- Algorithms Design and Analysis, Udit Agarwal, Dhanpat Rai

6. Course Outcomes:

After going through this course the Students will be able to:

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS491.1	Implement Divide and Conquer algorithms for searching, sorting problems.	Apply	L3
CS491.2	Apply Greedy algorithms to solve optimization problems	Apply	L3
CS491.3	Apply Dynamic Programming to solve optimization problems	Apply	L3
CS491.4	Develop solutions using Graph algorithms for shortest path and traversal problems.	Develop	L5
CS491.5	Solve complex computational problems using Backtracking and Branch-and-Bound techniques.	Design	L6

7. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5
1	3	-	-	-	-
2	-	3	-	-	-
3	-	-	3	-	-
4	-	-	-	3	-
5	-	-	-	-	3

8. Mapping of the Course outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	3	2	2	-	-	-	-	-	2
CO2	3	3	3	2	3	-	-	-	-	-	2
CO3	3	3	3	2	3	-	-	-	-	-	2
CO4	3	3	3	3	3	2	-	-	-	-	2
CO5	3	3	3	3	2	-	-	-	-	-	3
AVG.	**3.0**	**2.8**	**3.0**	**2.4**	**2.6**	**2.0**	**-**	**-**	**-**	**-**	**2.2**

9. Mapping to PSO

Course Outcome	PSO1	PSO2	PSO3
CS491.CO1	3	2	—
CS491.CO2	3	2	—
CS491.CO3	3	2	—
CS491.CO4	3	2	1
CS491.CO5	3	2	1

***** End of Syllabus*****

Course Name: Object Oriented Programming Lab
Course Code: CS492
(Semester – IV)
Category: Major
Course Broad Category: Engineering Science Course
[For Computer Science and Engineering]

1. **Course Prerequisite:**
   Before enrolling in this lab, students should have:
   - Basic knowledge of procedural programming (C, loops, functions).
   - Understanding of data structures (arrays, linked lists).
   - Basic familiarity with an Integrated Development Environment (IDE) like Eclipse, NetBeans, or Code::Blocks.
   - Fundamental knowledge of memory management and object-oriented concepts.

2. **Course Learning Objectives:**
   After successful completion of this lab, students will be able to:
   i. **Implement** object-oriented programming principles such as **classes, constructors, method overloading, inheritance, and method overriding** in software development.
   ii. **Develop** modular and reusable code by utilizing **wrapper classes, arrays, interfaces (multiple inheritance, extending interfaces), and packages**.
   iii. **Apply** advanced OOP techniques, including **multithreading and applet programming**, to build interactive and efficient applications.
   iv. **Demonstrate** proficiency in **graphical user interface (GUI) programming** using modern development tools like **Swing, JavaFX, or similar frameworks in C++.**

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies.
   Evaluation System –
   For each laboratory courses there shall be a CIA during the semester for 60 marks and 40 marks for ESE.
   - **Continuous Internal Assessment of laboratory courses (PCIA) [60 Marks]**
     - Regular Laboratory Performance (20 Marks)
     - Experiment Execution and Results (15 Marks)
     - Laboratory Record/Report (10 Marks)
     - Viva Voce/Quiz (5 Marks)
     - Attendance (10 Marks)
   - **End Semester Examination of Laboratory Course (PESE) [40 marks]**
     - Write-Up about the given Experiment/Program [10 marks]
     - Execution of the Experiment/Program [10 marks]
     - Evaluation of the outcome or results of the Experiment/Program [10 marks]
     - Viva-Voce (on the concerned Experiment/Programming) [10 marks]

4. **Course Content:**
   Course Name: Object Oriented Programming Lab
   Course Code: CS492
   Hours per Week: 4P: 0T: 0P
   Credits: 2

Module	Topics : Lab Experiments
Unit 1:	Assignments on simple Java programming using operators, control statements & loops, array, constructor, Assignments on Command line arguments, Programming using keyboard input by Scanner classes
Unit 2:	Assignments on Basic String handling & I/O, Programming using keyboard input by implementing Buffered Reader, class, object, and method, access specifier, method/constructor overloading, inheritance, overriding, Wrapper class.
Unit 3:	Assignments on developing interfaces- multiple inheritance, extending interfaces, dynamic method dispatch, abstract classes & methods
Unit 4:	Assignments on creating and accessing packages, Programming on exception handling using try-catch block, implementing throw and throws keywords, using finally block, creating user-defined exception
Unit 5:	Assignments on creating child threads i) by extending thread class ii) by implementing runnable interface, creating child threads by assigning thread priorities. Assignments on multithreaded programming

Unit 6:	Assignments on Programming on creating simple applet to display some message, creating applet two add 2 integers, creating applet to do GUI based programming, Assignments on Java Swing, Some Case Studies (e.g. data entry System / application, drawing an image on canvas.)

Note: Use Java for programming
Any experiment specially designed by the college (Detailed instructions for Laboratory Manual to be followed for further guidance)

6. References:

Text & References Books:

- Herbert Schildt, *Java: The Complete Reference*, McGraw Hill.
- Kathy Sierra & Bert Bates, *Head First Java*, O'Reilly Media.
- Erich Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

7. Course Outcomes:

After going through this course the Students will be able to:

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS492.1	**Apply** the concepts of class, constructor, overloading, inheritance, and overriding to develop Java programs.	**Apply**	L3
CS492.2	**Implement** Java wrapper classes and arrays for effective data handling.	**Implement**	L3
CS492.3	**Develop** programs using interfaces, multiple inheritance, and extending interfaces.	**Develop**	L6
CS492.4	**Design** and access Java packages for modular programming.	**Design**	L6
CS492.5	**Develop** multithreaded programs to achieve concurrency and synchronization.	**Develop**	L6
CS492.6	**Design and implement** Java applets for interactive web applications.	**Design and implement**	L6

8. Mapping of course outcomes to module / course content

Module No.	Topics Covered	Mapped COs
1	Classes, Constructors, Method Overloading, Inheritance, and Overriding	CO1
2	Wrapper Classes, Arrays	CO2
3	Developing Interfaces (Multiple Inheritance, Extending Interfaces)	CO3
4	Creating and Accessing Packages	CO4
5	Multithreaded Programming	CO5
6	Applet Programming, GUI Development	CO6

9. Mapping of the Course outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	3	2	2	-	-	-	1	-	1
CO2	3	2	2	2	2	-	-	-	-	-	1
CO3	3	2	3	2	2	-	-	1	1	-	1
CO4	2	1	3	2	2	-	-	-	1	-	1
CO5	3	2	3	2	3	-	-	1	2	-	2
CO6	3	1	2	1	2	-	-	1	2	-	2
AVG.	**2.83**	**1.66**	**2.66**	**1.83**	**2.16**	**-**	**-**	**0.5**	**1.16**	**-**	**1.33**

10. Mapping to PSO

Course Outcome	PSO1	PSO2	PSO3
CS492.CO1	2	3	—
CS492.CO2	2	3	—
CS492.CO3	2	3	—
CS492.CO4	2	3	—
CS492.CO5	2	3	1
CS492.CO6	1	2	—

***** End of Syllabus*****

Course Name: Database Management System Lab
Course Code: CS493
(Semester – IV)
Category: Major
Course Broad Category: Engineering Science Course
[For Computer Science and Engineering]

1. **Course Prerequisite:**
   Basics of data structure and programming concepts

2. **Course Learning Objectives:**
   i. Learn to design, create, and maintain databases.
   ii. Develop SQL skills for querying and managing data.
   iii. Understand normalization for efficient database design.
   iv. Explore transaction management and ACID properties.
   v. Practice database programming with real-world applications**.**

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies.
   Evaluation System –
   For each laboratory courses there shall be a CIA during the semester for 60 marks and 40 marks for ESE.
   - **Continuous Internal Assessment of laboratory courses (PCIA) [60 Marks]**
     - Regular Laboratory Performance (20 Marks)
     - Experiment Execution and Results (15 Marks)
     - Laboratory Record/Report (10 Marks)
     - Viva Voce/Quiz (5 Marks)
     - Attendance (10 Marks)
   - **End Semester Examination of Laboratory Course (PESE) [40 marks]**
     - Write-Up about the given Experiment/Program [10 marks]
     - Execution of the Experiment/Program [10 marks]
     - Evaluation of the outcome or results of the Experiment/Program [10 marks]
     - Viva-Voce (on the concerned Experiment/Programming) [10 marks]

4. **Course Content:**
   Course Name: Database Management System Lab
   Course Code: CS493
   Hours per Week: 0L : 0T : 4P
   Credits: 2

Module	Topics : Laboratory Experiments
Unit 1: Creating Database	• Creating a Database • Creating a Table • Specifying Relational Data Types • Specifying Constraints • Creating Indexes
Unit 2: Table and Record Handling	• INSERT statement • Using SELECT and INSERT together • DELETE, UPDATE, TRUNCATE statements • DROP, ALTER statements
Unit 3: Retrieving Data from a Database	• The SELECT statement • Using the WHERE clause • Using Logical Operators in the WHERE clause • Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING Clause • Using Aggregate Functions • Combining Tables Using JOINS • Subqueries
Unit 4: Database Management	• Creating Views • Creating Column Aliases • Creating Database Users • Using GRANT and REVOKE

Unit 5:	• Cursors in Oracle PL / SQL • Writing Oracle PL / SQL Stored Procedures • Creating and Using Triggers in Oracle PL/SQL

Any experiment specially designed by the college (Detailed instructions for Laboratory Manual to be followed for further guidance)

6. References:

Text & References Books:

- SQL, PL/SQL – The Programming Language of Oracle, by Ivan Bayross
- Getting Started with SQL: A Hands-On Approach for Beginners, by Thomas Nield
- Practical SQL, by Anthony Debarros
- Learning SQL: Generate, Manipulate, and Retrieve Data, by Alan Beaulieu
- SQL The Complete Reference, by James Groff, Paul Weinberg and Andy OppelCourse Outcomes:

7. Course Outcomes:

After going through this course the Students will be able to:

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
CS493.1	Analyze and transform an Entity Relationship Model in a relational database schema and to use data definition language to implement the schema using a DBMS	Analyze	L4
CS493.2	Declare and enforce integrity constraints on a database using a DBMS	Declare	L1
CS493.3	Populate and query a database using SQL DML/DDL commands.	Populate	L3
CS493.4	Retrieve data from a database.	Retrieve	L3
CS493.5	Describe and implement relational algebra expression using aggregate functions, joins and subqueries.	Describe	L1
CS493.6	Compile programs in PL/SQL including stored procedures, stored functions, cursors, packages.	Compile	L6

8. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1	Y	Y	Y	-	-	-
2	-	-	Y	Y	-	-
3	-	-	Y	Y	Y	-
4	Y	Y	Y	Y	Y	-
5	-	-	-	-	-	Y

9. Mapping of the Course outcomes to Program Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	3	1	3	1	1	2	-	1	2
CO2	3	2	3	2	3	1	-	2	-	1	2
CO3	3	2	3	2	3	1	2	2	-	1	2
CO4	3	2	3	2	3	1	2	2	-	1	2
CO5	3	2	3	3	3	1	2	2	-	1	2
CO6	3	2	3	2	3	1	2	2	-	1	2
AVG.	**3**	**2**	**3**	**3**	**3**	**1**	**1.8**	**2**	**0**	**1**	**2**

10. Mapping to PSO

Course Outcome	PSO1	PSO2	PSO3
CS493.CO1	3	2	—
CS493.CO2	2	2	—
CS493.CO3	3	3	—
CS493.CO4	2	3	—
CS493.CO5	3	3	—
CS493.CO6	2	3	1

***** End of Syllabus*****